

API Doc

Aperçu

Cette API offre la possibilité de créer, lire, modifier et rechercher et supprimer (CRUD) des utilisateurs dans l'annuaire du Utilisateur.

Transport

HTTP obligatoire

Style de l'interface

REST/JSON

Authentification

L'authentification se fait via un login et un mot de passe en utilisant le mode d'authentification HTTP Basic.

Contrôle d'accès

En utilisant la fonctionnalité des rôles affectés aux utilisateurs techniques il est possible d'autoriser ou non les différentes actions: créer, modifier, rechercher, supprimer.

Liste/recherche un Utilisateur

URL	Méthode	Contenu	Paramètres
/api/users/	GET	document JSON	voir plus loin

Paramètres

Nom	Sémantique	Valeurs permises	Exemple
ordering	Permet de passer le nom d'un champ par lequel les réponses seront ordonnées, préfixé par - l'ordre est inversé, ex.: ordering=last_name	[-]date_joined, [-]modified, [-]first_name, [-]last_name	
first_name	recherche exacte sur le prénom pour une valeur exacte		
first_name__iexact	recherche exacte sur le prénom pour une valeur exacte en ignorant la casse		
first_name__icontains	recherche exacte sur le prénom pour une sous chaîne en ignorant la casse		
first_name__gte	recherche sur le prénom supérieure ou égale à la valeur		
first_name__lte	recherche sur le prénom inférieure ou égale à la valeur		
first_name__gt	recherche sur le prénom supérieure à la valeur		
first_name__lt	recherche sur le prénom inférieure à la valeur		
last_name + __iexact, __icontains, __gte, __lte, __gt,	idem que pour le prénom mais avec le nom		

__lt			
modified__gte	recherche sur la date de dernière modification supérieure ou égale à la valeur	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS	curl -X GET -u partenaire:geronimo https://connexion-demo.dev.entrouvert.org/api/users/?modified__gte=2017-03-28T00:00:00
modified__lte	recherche sur la date de dernière modification inférieure ou égale à la valeur	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS	
modified__gt	recherche sur la date de dernière modification supérieure à la valeur	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS	
modified__lt	recherche sur la date de dernière modification inférieure à la valeur	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS	
email	recherche sur le courriel exact		
email__iexact	recherche sur le courriel exact en ignorant la casse		

Format retour

Le document retour est un objet/dictionnaire JSON, les résultats sont toujours paginés, chaque page a une taille de 100, le lien vers la page suivante est dans la propriété next et celui vers la page précédente dans la propriété previous. La propriété results contient la liste des résultats.

Les résultats contiennent les propriétés suivantes:

Propriété	Description	Format
uuid	identifiant unique	
username	nom d'utilisateur	chaîne quelconque
password	mot de passe chiffré	
date_joined	date d'inscription	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS
email	courriel	courriel
email_verified		booléen
first_name	prénom	chaîne quelconque
last_name	nom	chaîne quelconque
is_active		booléen
is_staff		booléen
is_admin		booléen
date_joined	date de dernière connexion	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS
modified	date de dernière modification	date et heure au format ISO-8601: YYYY-MM-DDTHH:MM:SS
ou	Collectivité de l'utilisateur	chaîne quelconque

Exemple cas-passant

```
GET /api/users/ HTTP/1.1
Authorization: Basic xxx
```

```
200 OK
Content-Type: application/json
Content-Length: xxxx
```

```

{
  "next": null,
  "previous": null,
  "results": [
    {
      "date_joined": "2017-08-21T14:28:00Z",
      "email": "john.doe@example.com",
      "email_verified": false,
      "first_name": "John Kevin Eric",
      "id": 4,
      "is_active": true,
      "is_staff": false,
      "is_superuser": false,
      "last_login": null,
      "last_name": "Doe",
      "modified": "2017-08-21T14:47:34.102909Z",
      "ou": "default",
      "password": "pbkdf2_sha256$20000$crynWolm1hVD$SpckEw0W6HNjDdex6RcKrGPB5N5eBwjCRT+KaFWF
ATA=",
      "username": "jdoe",
      "uuid": "73e62dc8de25486e901753d127309eef"
    }
  ]
}

```

Cas non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (paramètre interdit ou avec une valeur invalide)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échoué	
403	Permission non accordée d'effectuer l'action	

Création d'un Utilisateur

URL	Méthode	Contenu	Paramètres
/api/users/	POST	document JSON décrivant le Utilisateur à créer (à spécifier)	non

Format entrée

Le document en entrée est un objet/dictionnaire JSON avec les propriétés suivantes:

Propriété	Description	Valeur requise	Valeur permise
username	nom d'utilisateur	non	chaîne quelconque
password	mot de passe	non	chaîne quelconque
first_name	prénom	oui	chaîne quelconque
last_name	nom	oui	chaîne quelconque
email	courriel	non	courriel

Les valeurs null ne sont pas autorisées.

Exemple

```
POST /api/users/ HTTP/1.1
Content-Type: application/json
Authorization: Basic xxxxx
Content-Length: xxx
```

```
{
  "email": "john.doe@example.com",
  "first_name": "John",
  "last_name": "Doe",
  "email": "john.doe@example.com",
  "username": "jdoe",
  "password": "toto"
}
```

Retour passant

```
201 Created
Content-Type: application/json
Content-Length: xxx
```

```
{
  "date_joined": "2017-08-21T14:28:00.194032Z",
  "email": "",
  "email_verified": false,
  "first_name": "John",
  "id": 4,
  "is_active": true,
  "is_staff": false,
  "is_superuser": false,
  "last_login": null,
  "last_name": "Doe",
  "modified": "2017-08-21T14:28:00.470957Z",
  "ou": "default",
  "password": "pbkdf2_sha256$20000$crynWolm1hVD$SpckEw0W6HNjDdex6RcKrGPB5N5eBwjCRT+KaFWFATA=",
  "username": "jdoe",
  "uuid": "73e62dc8de25486e901753d127309eef"
}
```

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant, etc..)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échoué	
403	Permission non accordée d'effectuer l'action	

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: xxxxx
```

```
{
  "errors": {
    "last_name": [
      "Ce champ est obligatoire."
    ]
  },
  "result": 0
}
```

Lecture d'un Utilisateur

URL	Méthode	Contenu	Paramètres
/api/users/<uuid>/	GET	document JSON décrivant le Utilisateur lu	non

```
GET /api/users/<uuid>/ HTTP/1.1
Authorization: Basic xxxxx
```

Retour passant

Code	Signification	Contenu
200	Un document JSON décrivant le Utilisateur	

```
200 Ok
Content-Type: application/json
Content-Length: xxxxx
```

```
{
  "date_joined": "2017-08-21T14:28:00Z",
  "email": "john.doe@example.com",
  "email_verified": false,
  "first_name": "John",
  "id": 4,
  "is_active": true,
  "is_staff": false,
  "is_superuser": false,
  "last_login": null,
  "last_name": "Doe",
  "modified": "2017-08-21T14:34:15.669381Z",
  "ou": "default",
  "password": "pbkdf2_sha256$20000$crynWolm1hVD$SpKew0W6HNjDdex6RcKrGPB5N5eBwjCRT+KaFWFATA=",
  "username": "jdoe",
  "uuid": "73e62dc8de25486e901753d127309eef"
}
```

Les propriétés sans valeur sont à null.

Retour non-passant

Code	Signification	Contenu
401	L'authentification a échouée	
403	Permission non accordée d'effectuer l'action	
404	L'id Utilisateur donné n'existe pas	

voir plus haut

Modification d'un Utilisateur

URL	Méthode	Contenu	Paramètres
/api/users/<uuid>/	PUT	document JSON décrivant le Utilisateur	non
/api/users/<uuid>/	PATCH	document JSON décrivant le Utilisateur	non

```
PATCH /api/users/<uuid>/ HTTP/1.1
Content-Type: application/json
```

Authorization: Basic xxx

Content-Length: xxx

```
{
  "first_name": "John Kevin"
}
```

L'action PUT nécessite que first_name et last_name (obligatoires) soient dans le document JSON. S'ils ne sont pas renseignés, une erreur 400 est retournée.

Retour passant

Code	Signification	Contenu
200	Ok	Un document JSON décrivant le Utilisateur modifié

200 Ok

Content-Type: application/json

Content-Length: xxx

```
{
  "date_joined": "2017-08-21T14:28:00Z",
  "email": "john.doe@example.com",
  "email_verified": false,
  "first_name": "John Kevin",
  "id": 4,
  "is_active": true,
  "is_staff": false,
  "is_superuser": false,
  "last_login": null,
  "last_name": "Doe",
  "modified": "2017-08-21T14:34:15.669381Z",
  "ou": "default",
  "password": "pbkdf2_sha256$20000$crynWolm1hVD$SpCKEw0W6HNjDdex6RcKrGPB5N5eBwjCRT+KaFWFATA=",
  "username": "jdoe",
  "uuid": "73e62dc8de25486e901753d127309eef"
}
```

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant, etc..)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échoué	
403	Permission non accordée d'effectuer l'action	
404	L'id Utilisateur donné n'existe pas	

HTTP/1.1 400 BAD REQUEST

Content-Type: application/json

Content-Length: xxxx

```
{
  "errors": {
    "last_name": [
      "Ce champ est obligatoire."
    ]
  },
  "result": 0
}
```

Suppression d'un Utilisateur

URL	Méthode	Contenu	Paramètres
/api/users/<uuid>/	DELETE	aucun	non

DELETE /api/users/<uuid>/ HTTP/1.1

Authorization: Basic xxx

Retour passant

Code	Signification	Contenu
204	Ok	aucun

204 No content

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant, etc..)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échouée	
403	Permission non accordée d'effectuer l'action	
404	L'id Utilisateur donné n'existe pas	JSON d'erreur

Forcer le changement de mot de passe à la prochaine connexion

URL	Méthode	Contenu	Paramètres
/api/users/<uuid>/force-password-reset/	POST	aucun	non

POST /api/users/<uuid>/force-password-reset/ HTTP/1.1

Authorization: Basic xxx

Retour passant

Code	Signification	Contenu
204	Ok	aucun

204 No content

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant, etc..)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échouée	
403	Permission non accordée d'effectuer l'action	
404	L'id Utilisateur donné n'existe pas	JSON d'erreur

Tests d'existence des comptes en masse

URL	Méthode	Contenu	Paramètres
/api/users/synchronization/	POST	Document JSON contenant la liste des ID Utilisateur à tester	non

```
POST /api/users/synchronization/ HTTP/1.1
Authorization: Basic xxx
Content-Type: application/json
Content-Length: xxxx
```

```
{
  "known_uuids":["73e62dc8de25486e901753d127309eef", "1234567890"],
}
```

Retour passant

```
200 Ok
Content-Type: application/json
Content-Length: xxxx
```

```
{
  "unknown_uuids":["1234567890"],
  "result":1
}
```

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant les erreurs rencontrées)	
401	L'authentification a échoué	
403	Permission non accordée d'effectuer l'action	

```
400 BadRequest
Content-Type: application/json
Content-Length: xxxx
```

```
{
  "detail":"JSON parse error - Expecting ',' delimiter: line 1 column 53 (char 52)"
}
```

Envoyer un jeton pour réinitialisation de mot de passe

URL	Méthode	Contenu	Paramètres
/api/users/<uuid>/password-reset/	POST	aucun	non

```
POST /api/users/<uuid>/password-reset/ HTTP/1.1
Authorization: Basic xxx
```

Retour passant

Code	Signification	Contenu

204	Ok	aucun
-----	----	-------

204 No content

Retour non-passant

Code	Signification	Contenu
400	Le format de la requête est invalide (JSON mal formaté, attribut manquant, etc..)	Un document JSON décrivant les erreurs rencontrées
401	L'authentification a échouée	
403	Permission non accordée d'effectuer l'action	
404	L'id Utilisateur donné n'existe pas	JSON d'erreur
500	L'utilisateur identifié par l'id Utilisateur n'a pas de courriel	JSON d'erreur